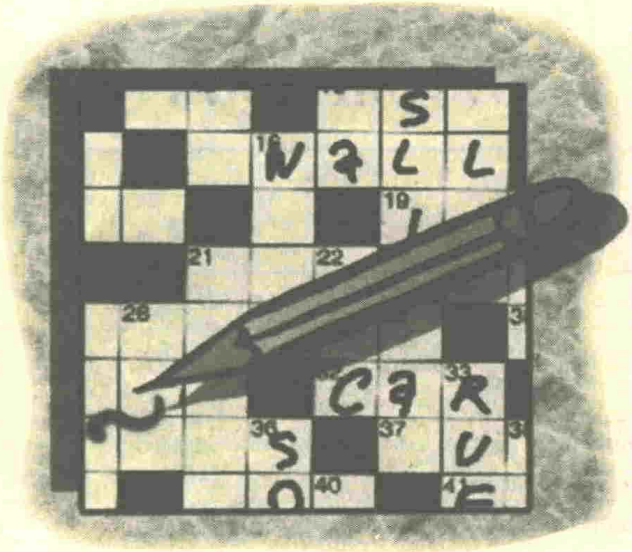
บุญเลิศ เอี่ยมทัศนา

โปรแกรมด้วย Delphi : กรอบตาราง

วิธีการนำกรอบตารางมาใช้ อย่างเช่นโปรแกรม CalGrid และยังใช้ร่วมกับการคำนวณ ในลักษณะของสเปรดชีตได้อีกด้วย



การใช้งานกรอบตารางก็คือใช้งานอย่างตาราง เช่น ทำเป็น ตารางสอน ตารางการนัดหมาย ตารางรายการสินค้า เป็นต้น และยังใช้ร่วมกับการคำนวณในลักษณะของสเปรดชีตได้อีกด้วย

ท้าวความ

ผมได้นำเรื่องของการโปรแกรมด้วยเดลไฟมากล่าวในหลายตอนแล้ว ส่วนใหญ่จะเป็นการนำคอมโปเนนต์มาแนะนำ ดังเช่นในตอนนี้จะเป็นเรื่องของกรอบตารางคือ StringGrid กับ DrawGrid เป็นคอมโปเนนต์สุดท้ายของเดลไฟ 1.0 ต่อไปจะเป็นของเดลไฟ 2.0 กับการโปรแกรมโดยทั่วไปครับ

กรอบตาราง

ในเดลไฟได้เตรียมกรอบตารางไว้ 2 ชนิดคือ StringGrid กับ DrawGrid (อยู่ในหน้า Additional) ซึ่งทั้งชื่อและภาพคอมโปเนนต์อาจทำให้เข้าใจผิดอย่างตรงข้ามคือ DrawGrid จะไม่วาดคือไม่แสดงภาพทั้งนั้น ไม่ว่าจะเป็นภาพรูปภาพหรือข้อความ เราจะต้องเป็นผู้จัดการแสดงเอง ส่วน StringGrid จะแสดงแค่ภาพข้อความและให้ผู้ใช้ป้อนข้อความได้

การให้ค่า

การให้ค่าแก่กรอบตารางกระทำได้ 2 วิธีคือ ให้จากค่าคงตัว หรือโหลดมาจากไฟล์ ดังจะแสดงการใช้กรอบรายการทั้งสอง ดังนี้

- StringGrid ในโปรแกรม SimGrid และ EdtGrid เพื่อแสดงการให้ค่าจากค่าคงตัวและจากไฟล์ตามลำดับ กับโปรแกรม CalGrid เพื่อแสดงการคำนวณเป็นโรว์และคอลัมน์
- DrawGrid ในโปรแกรม BrwsIco เพื่อแสดงภาพข้อความและรูปภาพที่โหลดมาจากไฟล์ ในฉบับหน้าครับ

โปรแกรม SimGrid

เพื่อแสดงการให้ค่าแก่ StringGrid ด้วยค่าคงตัว ดังผลการรันในรูปที่ 1

เตรียมโปรแกรม

จะเตรียมโปรแกรมได้ดังนี้

1. เปิดโปรเจ็กต์ใหม่ แล้วให้ค่า Font ของ Form1 ดังนี้
   - Name ให้เป็น MS Sans Serif
   - Size ให้เป็น 8

```
type  Str19T = String[19];
const  ColNme : array[0..5] of Str19T =
        (`    Menu', 'Monday', 'Tuesday',
         'Wednesday', 'Thursday', 'Friday');
       RowNme : array[1..3] of Str19T =
        ( `  Main', `  Soup', `  Fruit');

const  FoodMnu : array[0..3, 0..5] of Str19T =
        ((`    Menu', 'Monday', 'Tuesday',
            'Wednesday', 'Thursday', 'Friday'),
         (`  Main', 'Meat', 'Chicken', 'Fish',
                'Pork', 'Lobster'),
         (`  Soup', 'Mushroom', 'Tomato', 'Thick cream',
                'Onion', 'Cold soup'),
         (`  Fruit', 'Banana', 'Durion', 'Orange',
                'Mango', 'Pineapple'));

procedure TForm1.Button1Click(Sender: TObject);
var  cl,rw : Integer;
begin
  for  cl := 0 to 5  do
    StringGrid1.Cells[cl, 0] := ColNme[cl];
  for  rw := 1 to 3  do
    StringGrid1.Cells[0, rw] := RowNme[rw];
end;

procedure TForm1.Button2Click(Sender: TObject);
var  cl,rw : Integer;
begin
   for  rw := 0 to 3  do
     for  cl := 0 to 5  do
       StringGrid1.Cells[cl, rw] := FoodMnu[rw, cl];
end;
```

ลิสติงที่ 1

2. กำหนดคอมโปเนนต์เป็น StringGrid1 ให้ค่าพรอพเพอร์ตี้ส์ดังนี้

- ColCount ให้เป็น 6 ส่วน RowCount ให้เป็น 4
- ให้ค่าแฟล็กของ Options คือ goRowSizing และ goColSizing ให้เป็น True เพื่อให้ผู้ใช้สามารถจัดขนาดของโรว์และคอลัมน์ได้

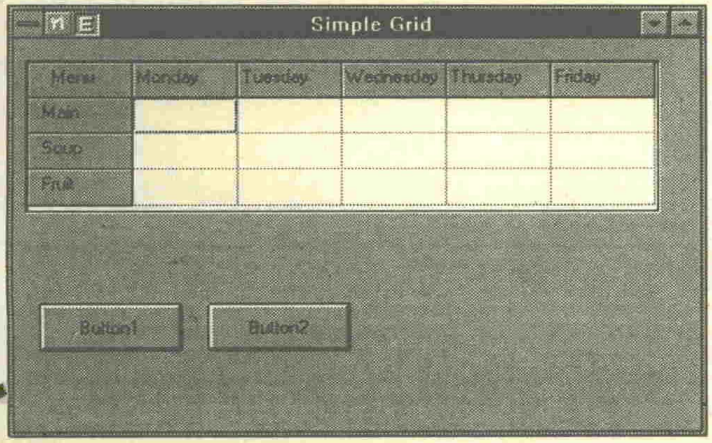
3. กำหนดคอมโปเนนต์เป็น Button1 และ Button2

4. ป้อนโปรแกรมตามลิสติงที่ 1

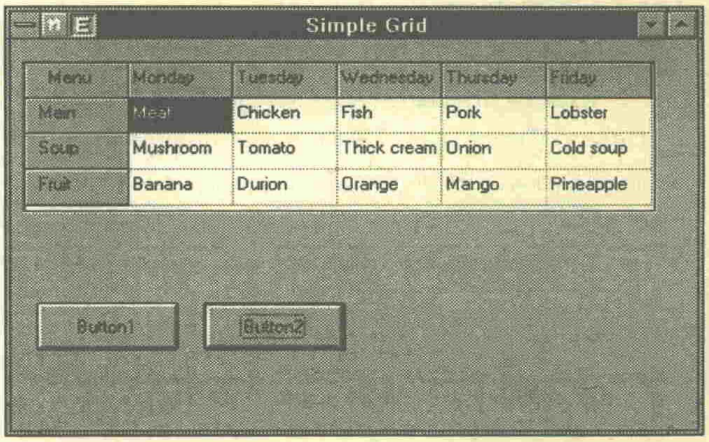
คำอธิบาย

1. ผลการรันตามรูปที่ 1 เกิดจากเลือกปุ่ม Button1 ซึ่งในกิจกรรม Button1Click ได้นำค่าใน ColNme และ RowNme ที่ได้ให้ค่าคงตัวไว้เป็นอะเรย์มิติเดี่ยว มาให้เป็นชื่อคอลัมน์และชื่อโรว์

2. ถ้าเลือกปุ่ม Button2 จะได้ผลดังรูปที่ 2 ซึ่งในกิจกรรม Button2Click จะนำค่าใน FoodMemu ที่ให้ค่าไว้เป็นอะเรย์สองมิติ มาให้ค่าแก่กรอบตารางใหม่ทั้งหมด



รูปที่ 1 ผลการรันโปรแกรม SimGrid หลังจากเลือกปุ่ม Button1



รูปที่ 2 หน้าต่างของโปรแกรม SimGrid หลังจากเลือกปุ่ม Button2

ลิสติงที่ 2

```
      Menu
Monday
Tuesday
Wednesday
Thursday
Friday
  Main
Meat
Chicken
Fish
Pork
Lobster
  Soup
Mushroom
Tomato
Thick cream
Onion
Cold soup
  Fruit
Banana
Durion
Orange
Mango
Pineapple
```

3. ในโปรแกรมจริงควรให้ค่าใน FormCreate ซึ่งจะเหมาะกับตารางที่ไม่เปลี่ยนแปลง

โปรแกรม EdtGrid

แต่ค่ารายการคือค่าที่แสดงในแต่ละช่องโดยเฉพาะส่วนที่เป็นพื้นขาว ค่านี้อาจเปลี่ยนแปลงได้เช่นในตารางสอนหรือในตารางการนัดหมาย ซึ่งการเปลี่ยนแปลงนี้ควรให้ผู้ใช้ป้อนแก้ได้ เช่น อาจให้ป้อนด้วย NotePad ดังเช่นในลิสติงที่ 2 เก็บไว้ในไฟล์ .txt แล้วโหลดมาให้ค่าแก่กรอบตารางใน FormCreate (จะได้ผลเช่นเดียวกับในรูปที่ 2)

อีกวิธีหนึ่งซึ่งคิดว่าน่าจะดีกว่าคือให้ผู้ใช้ป้อนแก้ในกรอบตารางแล้วเก็บลงไฟล์ ตามที่จะแสดงในโปรแกรมนี้ กล่าวคือ

เตรียมโปรแกรม

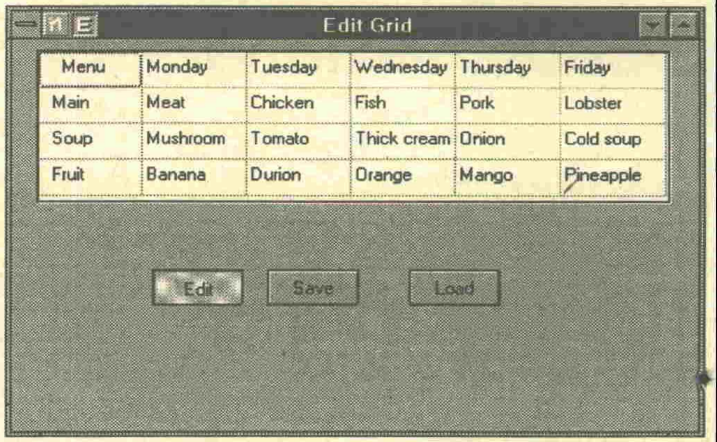
จะเตรียมโปรแกรมตามรูปที่ 3 ได้ดังนี้

1. เปิดฟอร์มใหม่และกำหนดคอมโปเนนต์เป็น StringGrid1 ให้ค่าพรอพเพอร์ตี้ส์เช่นเดียวกับในโปรแกรม SimGrid โดยเพิ่มการให้ค่าแฟล็กใน Options ของ StringGrid1 คือ goEditing และ goTab เป็น True เพื่อให้ผู้ใช้สามารถป้อนข้อความเป็นค่ารายการและใช้คีย์ Tab ได้

2. กำหนดคอมโปเนนต์เป็น SpeedButton รวม 3 ปุ่ม แล้วให้ค่าพรอพเพอร์ตี้ส์ของแต่ละปุ่มจากซ้ายไปขวาดังนี้

- Caption ให้เป็น Edit, Save และ Load
- Name ให้เป็น spbEdit, spbSave และ spbLoad
- spbEdit.GroupIndex ให้เป็น 1

3. ป้อนโปรแกรมตามลิสติงที่ 3



รูปที่ 3 หน้าต่างของโปรแกรม EdtGrid หลังจากที่ผู้ใช้เลือกปุ่ม Edit และได้ป้อนข้อความที่เป็นค่ารายการแล้ว

ลิสติงที่ 3

```
const  foodFiNme = 'foodmenu.txt';

procedure TForm1.spbEditClick(Sender: TObject);
begin
 with  StringGrid1  do
  if  spbEdit.Down  then
    begin  FixedCols := 0;  FixedRows := 0  end
  else
    begin  FixedCols := 1;  FixedRows := 1  end;
end;

procedure TForm1.spbSaveClick(Sender: TObject);
var  oStrS : TStringList;  cl,rw : Integer;
begin
  oStrS := TStringList.Create;
  with  StringGrid1  do
    for rw := 0 to RowCount - 1 do
     for  cl := 0 to ColCount - 1 do
       oStrS.Add(Cells[cl,rw]);
  oStrS.SaveToFile(foodFiNme);
  oStrS.Free;
end;

procedure TForm1.spbLoadClick(Sender: TObject);
var  oStrS : TStringList;  cl,rw,itm : Integer;
begin
  oStrS := TStringList.Create;
  oStrS.LoadFromFile(foodFiNme);  itm := 0;
  for rw := 0 to StringGrid1.RowCount - 1 do
    for  cl := 0 to StringGrid1.ColCount - 1 do
     begin  StringGrid1.Cells[cl,rw] := oStrS[itm];
           inc(itm)  end;
  oStrS.Free;
end;
```

คำอธิบาย

1. เมื่อผู้ใช้ต้องการป้อนรายการโดยเฉพาะรายการที่เป็นชื่อคอลัมน์และชื่อโรว์ ให้เลือกปุ่ม Edit ซึ่งในโปรแกรมของกิจกรรม spbEditClick จะตรวจว่า

- ถ้าปุ่มนี้ขุบ จะให้ค่า FixedCols และ FixedRows เป็น 0 เท่ากับเป็นการยกเลิกส่วนที่เป็นชื่อคอลัมน์และชื่อโรว์ ทำให้ป้อนข้อความในสองส่วนนี้ได้ ซึ่งเมื่อผู้ใช้ป้อนข้อความทั้งหมดจะได้ผลดังรูปที่ 3 (ขอให้ป้อนข้อความเช่นตามนี้ด้วย)

- เมื่อเลือกปุ่ม Edit ซ้ำ จะให้มีส่วนทั้งสองนี้ใหม่ได้ผลดังรูปที่ 2

2. เมื่อผู้ใช้เลือกปุ่ม Save จะมาทำงานที่กิจกรรม spbSaveClick ซึ่งได้ให้กำเนิด oStrS เป็นออปเจ็กต์ของสกุล TStringList เพื่อเก็บรายการไลน์สตริง (คือสตริงที่ปิดท้ายด้วยอักขระ #13#10)

- นำค่ารายการในแต่ละช่องคือใน CellS ซึ่งเป็นไลน์สตริง มาเก็บใน oStrS ติดต่อกันไปเป็นบรรทัด ๆ

- ให้ oStrS เก็บลงไฟล์ เป็นเท็กซ์ไฟล์ (เมื่อให้ NotePad อ่านจึงได้ผลดังลิสติงที่ 2) เสร็จแล้วให้ขุบ oStrS

3. ถ้าผู้ใช้เลือกปุ่ม Load จะมาทำงานที่กิจกรรม spbLoadClick ซึ่งทำให้ oStrS อ่านข้อมูลจากไฟล์มาให้แก่กรอบตาราง จะได้ผลดังรูปที่ 2 เช่นกัน

4. ในโปรแกรมจริงควรได้เตรียมเท็กซ์ไฟล์ไว้ครั้งหนึ่ง และให้โหลดใน FormCreate ส่วนปุ่มโหลดอาจให้มีหรือไม่ตามความต้องการ

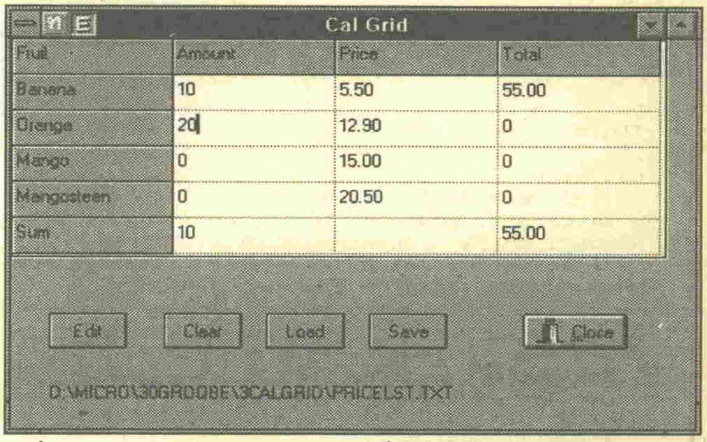
ปัญหาเรื่องไดเรคทอรี

การเตรียมโปรแกรมตามที่กล่าวมานี้ ถือว่าได้ดำเนินการตามที่ได้เคยกล่าวในฉบับก่อน ๆ คือ

- จะต้องเตรียมไดเรคทอรีเฉพาะของแต่ละโปรแกรม

- เมื่อเปิดโปรเจ็กต์ใหม่แล้วให้เก็บไฟล์ลงไดเรคทอรีของตน (File/Save Project As)

เมื่อดำเนินการตามนี้แล้ว เช่น ของโปรแกรม EdtGrid ก็ควรจะได้ไฟล์ foodmenu.txt ในไดเรคทอรีของโปรแกรมนี้



รูปที่ 4 ผลการรันโปรแกรม CalGrid เมื่อผู้ใช้ป้อนจำนวนส้มเป็น 20 แล้วแต่ยังไม่กดคีย์ Tab

แต่อาจมีปัญหาได้ หากผู้ใช้เปลี่ยนไดเรคทอรี เช่น ใช้ File Manager จะแสดงวิธีแก้ปัญหานี้ในกิจกรรม FormCreate ในโปรแกรม CalGrid

โปรแกรม CalGrid

ในกรณีที่ต้องการการคำนวณดังเช่นตามรูปที่ 4 เมื่อผู้ใช้ป้อนจำนวนผลไม้คือ 20 เป็นจำนวนของส้ม แล้วกดคีย์ Tab (ตามรูปยังไม่ได้กด) จะดำเนินการดังนี้

- นำจำนวน (Amount) คือ 20 ไปคูณกับราคา (Price) คือ 12.90 แล้วให้ผลคูณในช่อง Total (ดังที่เห็นของ Banana ได้ผลเป็น 55.00) เป็นราคาผลไม้รายการนั้น

ลิสติงที่ 4

```
PROGRAM  CalGrid;  { calgrid.dpr }

Form == Form1 ( Cal Grid )
Unit == CalGrid1  { calgrid1.pas }

Components :-
1 Additional :: StringGrid == StringGrid1
  - ColCount = 4
  - DefaultColWidth = 100
  - Options = [ goRowSizing(T),  goColSizing(T), goEditing(T), goTabs(T) ]
  - RowCount = 6
  - ScrollBar = ssNone
2 Additional :: SpeedButton == spbEdit ( Edit )
  - AllowAllUp = True
  - GroupIndex = 1
3 Additional :: SpeedButton == spbClear ( Clear )
4 Additional :: SpeedButton == spbLoad ( Load )
5 Additional :: SpeedButton == spbSave ( Save )
6 Additional :: BitBtn == BitBtn1
  - Kind = bkClose
```

→

ลิสติงที่ 4 (ต่อ)

```
(* --------- Declaration part --------- *)

type
  TForm1 = class(TForm)
    ..................
  private
    { Private declarations }
    txtChngF : Boolean;
    edtCl,edtRw : Integer;
    function  GetCellAsInt(cl,rw : Integer; def : LongInt) : LongInt;
    procedure SetIntCellAsStr(cl,rw : Integer; num : LongInt);
    function  GetCellAsReal(cl,rw : Integer; def : Real) : Real;
    procedure SetRealCellAsCrncyStr(cl,rw : Integer; num : Real);
    procedure CalRow(rw,resCol : Integer);
    procedure SumIntCol(cl,resRw : Integer);
    procedure SumRealCol(cl,resRw : Integer);
  public
    { Public declarations }
  end;

(* --------- Implementation part --------- *)

type    Str79T = String[79];
const   GridTxtFiNme : Str79T = 'pricelst.txt';
const   AmountColK = 1;   PriceColK = 2;   TotalColK = 3;
        SumRowK    = 5;


{$I loadsave.inc}

{ FileCreate('NEWFILE.TXT') }

procedure TForm1.FormCreate(Sender: TObject);
var  temNme : String[15];
begin
  temNme := GridTxtFiNme;
  GridTxtFiNme := ExpandFileName(temNme);
  if FileExists(GridTxtFiNme) then  spbLoadClick(Sender);
  txtChngF := False;
  Label1.Caption := GridTxtFiNme;
end;

procedure TForm1.spbClrClick(Sender: TObject);
var  rw : Integer;
begin
  for rw := 1 to StringGrid1.RowCount - 1 do
     StringGrid1.Cells[AmountColK,rw] := '0';
  for rw := 1 to StringGrid1.RowCount - 1 do
     StringGrid1.Cells[TotalColK,rw] := '0';
end;

procedure TForm1.StringGrid1SelectCell(Sender: TObject; Col, Row: Longint;
                                       var CanSelect: Boolean);
begin
   if  spbEdit.Down  then  Exit;
   if  Col = AmountColK  then  CanSelect := True
   else
     begin
     CanSelect := False;
     if  StringGrid1.Row < StringGrid1.RowCount-2  then
       StringGrid1.Row := StringGrid1.Row + 1;
     end;
end;

procedure TForm1.StringGrid1SetEditText(Sender: TObject;
                  ACol,ARow: Longint; const Value: String);
begin
  txtChngF := True;  edtCl := ACol;  edtRw := ARow;
end;

procedure TForm1.StringGrid1KeyUp(Sender: TObject; var Key: Word;
                                  Shift: TShiftState);
begin
```

→

- บวกยอดรวมค่าในคอลัมน์ Amount เว้นของ Sum นำไปให้ในช่องล่างสุด คือช่อง Amount ของ Sum เป็นจำนวนผลไม้รวม

- บวกรวมค่าในคอลัมน์ Total เว้นของ Sum นำไปใช้ในช่องล่างสุดเช่นกัน เป็นราคาผลไม้รวม

และให้ผู้ใช้สามารถอีดิตคือป้อนหรือป้อนแก้ข้อความได้ทั้งในส่วนประจำ (ชื่อคอลัมน์และชื่อโรว์) และส่วนข้อมูล (ส่วนที่เหลือ)

เตรียมโปรแกรม

เตรียมโปรแกรมตามรูปที่ 4 และตามลิสติงที่ 4 แต่ยังไม่ต้องป้อนโปรแกรมของกิจกรรม StringGrid1DrawCell โดยที่ในการป้อนโปรแกรมมีรายละเอียดตามที่จะกล่าวในหัวข้อต่อไป

การอินคลูดไฟล์

เนื่องจากต้องการนำโปรแกรมของกิจกรรม spbEditClick, spbSaveClick และ spbLoadClick ในโปรแกรม EdtGrid มาใช้ในโปรแกรม CalGrid ซึ่งอาจจะลอกมาไว้ในยูนิตของโปรแกรม CalGrid หรือใช้วิธีผนวกไฟล์ (include file) จะใช้วิธีหลังเพื่อแสดงวิธีผนวกไฟล์ของกิจกรรมอีเวนต์คือ

1. ให้ลอก (copy) โปรแกรมของกิจกรรมทั้งสามนี้ลงคลิปบอร์ด ปะ (paste) ให้แก่ NotePad แล้วเก็บลงไฟล์ โดยให้อยู่ในไดเรคทอรีเดียวกับโปรแกรม CalGrid ให้ชื่อไฟล์เป็น loadsave.inc

2. ผนวกเข้ากับโปรแกรม CalGrid โดยใช้ไดเรคทีฟ I ดังที่เห็นในลิสติงที่ 4

3. สร้างโครงโปรแกรมของปุ่ม Edit, Load และ Save ด้วยวิธีเดียวกับที่จะป้อนโปรแกรมของอีเวนต์ OnClick ให้แก่ปุ่มเหล่านี้ เช่น ดับเบิลคลิ๊กที่ปุ่มทั้งสาม

ลิสติงที่ 4 (ต่อ)

```
  if  (Key = vk_Tab) and (txtChngF = True)  then
   with  StringGrid1  do
    begin
    txtChngF := False;
    CalRow(edtRw, TotalColK);
    SumIntCol(AmountColK, SumRowK);
    SumRealCol(TotalColK, SumRowK);
    end;
end;

(* -------- *)

function  TForm1.GetCellAsReal(cl,rw : Integer; def : Real) : Real;
var  cde : Integer;
begin
  Val(StringGrid1.CellS[cl,rw], Result, cde);
  if  cde <> 0  then  Result := def;
end;

procedure TForm1.SetRealCellAsCrncyStr(cl,rw : Integer; num : Real);
var  txt : String[19];
begin
  Str(num:3:2, txt);
  StringGrid1.CellS[cl,rw] := txt;
end;

function  TForm1.GetCellAsInt(cl,rw : Integer; def : LongInt) : LongInt;
var  cde : Integer;
begin
  Val(StringGrid1.CellS[cl,rw], Result, cde);
  if  cde <> 0  then  Result := def;
end;

procedure TForm1.SetIntCellAsStr(cl,rw : Integer; num : LongInt);
begin
  StringGrid1.CellS[cl,rw] := IntToStr(num);
end;

(* -------- *)

procedure TForm1.CalRow(rw,resCol : Integer);
var  amnt : Integer;  prce,ttl : Real;
begin
  amnt := GetCellAsInt(AmountColK,rw, 0);
  prce := GetCellAsReal(PriceColK,rw, 0);
  ttl := amnt * prce;
  SetRealCellAsCrncyStr(resCol,rw, ttl);
end;

procedure TForm1.SumIntCol(cl,resRw : Integer);
var  isum : LongInt;  rw : Integer;
begin
  with  StringGrid1  do
    begin
    isum := 0;
    for  rw := 1 to resRw-1  do
      isum := isum + GetCellAsInt(cl,rw, 0);
    SetIntCellAsStr(cl,resRw, isum);
    end;
end;

procedure TForm1.SumRealCol(cl,resRw : Integer);
var  rsum : Real;  rw : Integer;
begin
  with  StringGrid1  do
    begin
    rsum := 0;
    for  rw := 1 to resRw-1  do
      rsum := rsum + GetCellAsReal(cl,rw, 0);
    SetRealCellAsCrncyStr(cl,resRw, rsum);
    end;
end;
```

→

นี้ (ครั้งละ 1 ปุ่ม) เมื่อได้โครงโปรแกรมแล้วให้ลบทิ้ง เพราะมีอยู่แล้วในไฟล์ loadsave.inc

คำอธิบาย

1. ใน FormCreate ได้ให้ดำเนินการดังนี้

- ให้ ExpandFileName หาชื่อไฟล์เต็ม (รวมไดเรคทอรี)

เพื่อแก้ปัญหาไดเรคทอรี แต่ก็ยังอาจมีปัญหาหากรันผ่านเดสไฟ ต้องรันกับโปรแกรมนี้โดยตรงเช่นที่ File Manager

- ตรวจว่าถ้ามีไฟล์ pricelst.txt แล้วให้โหลดมาใช้ ทำให้ไม่ต้องแบ่งการเตรียมโปรแกรมเป็น 2 ขั้นตอน

2. ได้แบ่งการทำงานออกเป็น 2 โหมดคือ

- โหมดอีดิต ผู้ใช้สามารถป้อนข้อความได้ทุกช่อง เช่น ให้ชื่อโรว์และคอลัมน์เมื่อเริ่มแรก หรือเมื่อต้องการเปลี่ยนชื่อผลไม้หรือราคา

- โหมดใช้งาน ผู้ใช้สามารถป้อนจำนวนผลไม้ในช่อง Amount ได้เท่านั้น

การบังคับให้เป็นตามนี้อยู่ในกิจกรรม StringGrid1SelectCell ซึ่งจะเกิดอีเวนต์นี้เมื่อผู้ใช้เลือกช่องใดช่องหนึ่ง เช่น คลิ๊กเม้าส์หรือกดคีย์ Tab จากช่องหนึ่งไปช่องหนึ่ง

ซึ่งจะอยู่ในโหมดอีดิต เมื่อผู้ใช้เลือกปุ่ม Edit จะทำให้ปุ่มนี้ยุบตัวให้ค่า Down เป็นจริงค้างอยู่ตามที่ให้ค่า GroupIndex ไม่ใช่ศูนย์ และจะยกตัว Down เป็นเท็จเมื่อเลือกซ้ำตามที่ให้ค่า AllowAllup เป็น True

3. ได้กำหนดไว้ว่าในโหมดใช้งาน เมื่อผู้ใช้ป้อนจำนวนผลไม้แล้ว จะต้องกดคีย์ Tab ถึงจะเป็นการป้อนที่สมบูรณ์ จึงจะคำนวณยอดต่าง ๆ ตามที่กล่าวมา ตามที่ได้จัดโปรแกรมไว้ดังนี้

ลิสติงที่ 4 (ต่อ)

```
procedure TForm1.StringGrid1DrawCell(Sender: TObject;
                          Col, Row: Longint;Rect: TRect;
                          State: TGridDrawState);
var  bufB : array[0..79] of Char;  algn : Word;
begin
  algn := 0;
  if  Col > 0   then  algn := dt_Right;
  if  Row = 0  then  algn := dt_Center;
  if  algn = 0  then  Exit;
  StrPCopy(bufB, StringGrid1.CellS[Col, Row]);
  if  Row = SumRowK  then
    begin
    StringGrid1.Canvas.Brush.Color := clGray;
    StringGrid1.Canvas.Font.Size := 10;
    StringGrid1.Canvas.Font.Color := clWhite;
    end;
  StringGrid1.Canvas.FillRect(Rect);
  Rect.Top := Rect.Top+2;  Rect.Right := Rect.Right-2;
  DrawText(StringGrid1.Canvas.Handle, bufB, -1, Rect, algn);
end;

end.
```

3.1 ใน StringGrid1SetEditText ซึ่งจะเกิดอีเวนต์นี้เมื่อผู้ใช้ป้อนค่าทุกคีย์ ได้ให้ txtChngF เป็นจริง เพื่อแสดงว่าผู้ใช้ได้ป้อนจำนวนผลไม้

3.2 ใน StringGrid1KeyUp ซึ่งจะมาทำงานที่กิจกรรมนี้ทุกครั้งที่ผู้ใช้ถอนมือจากคีย์เช่นกัน ได้ให้ตรวจว่าถ้าเป็นคีย์ Tab และได้ป้อนข้อมูล เพราะผู้ใช้อาจกดคีย์ Tab1 โดยไม่ป้อน จะให้

- CalRow คำนวณราคาผลไม้รายการนั้น
- SumIntCol บวกรวมจำนวนผลไม้รวม
- SumReal บวกรวมจำนวนผลไม้รวม

สำหรับโปรแกรมของกิจกรรมทั้งสามนี้ เป็นโปรแกรมที่ให้ผลตามที่กล่าวในตอนต้น จึงของดคำอธิบายครับ

4. สำหรับปุ่ม Close ซึ่งผมไม่ค่อยได้นำมาใช้ คือเพียงแต่กำหนดปุ่มนี้และให้ค่า Kind ตามคอมโปเนนต์แม็พ ก็จะได้ปุ่ม Close เทียบเท่ากับรายการ Close ในซิสเต็มเมนู โดยไม่ต้องป้อนโปรแกรม

หากต้องการแสดงผลเอง

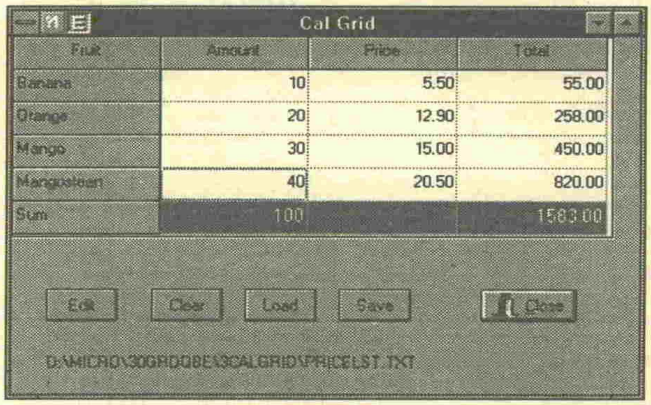
หากต้องการให้แสดงผลตามที่ต้องการดังรูปที่ 5 คือ

- แสดงชื่อคอลัมน์ตรงกึ่งกลาง
- แสดงตัวเลขชิดขวา
- ระบายสีโรว์ที่เป็นผลรวม และให้แสดงตัวเลขขนาดใหญ่กว่า

จะดำเนินการได้โดยจัดโปรแกรมสนองตอบอีเวนต์ OnDrawCell คือ StringGrid1DrawCell ดังแสดงในตอนท้ายของลิสติงที่ 4

โดยที่การแสดงภาพ เลือกได้ 2 วิธีคือ แสดงเองทั้งหมด (ให้ DefaultDrawing เป็น False) หรือให้ StringGrid1 แสดงด้วย (ให้ DefaultDrawing เป็น True) ของโปรแกรมนี้ให้ StringGrid1 แสดงร่วมด้วย โดยให้แสดงชื่อโรว์

ซึ่งความจริงควรแสดงเองทั้งหมดเพราะ



รูปที่ 5 กรอบตารางของโปรแกรม CalGrid เมื่อให้แสดงภาพด้วยโปรแกรมของเรา

- เมื่อให้ StringGrid1 แสดงภาพ จะแสดงทั้งหมดก่อน ส่วนของเราจะเป็นการแสดงทับ
- เขียนโปรแกรมได้ง่ายกว่า
- จะได้บรรทัด Sum เป็นรูปแบบเดียวกัน โดยไม่ต้องเขียนโปรแกรมเพิ่ม

ท้ายตอน

คงจะพอให้มองเห็นวิธีการนำกรอบตารางมาใช้ อย่างเช่น โปรแกรม CalGrid สามารถปรับปรุงใช้ติดตามการเลือกตั้งได้เป็นอย่างดี และคงทัน

ในฉบับหน้าจะเป็นเรื่องของ DrawGrid ประกอบกับเรื่องอื่นอีกด้วย และถ้าผมได้รับ Delphi 2.0 ก็จะนำมากล่าวร่วมด้วย